

## **Integrating Cobra for Cost/Earned Value Management in the Oracle Financials Environment.**

J. Steven Joines, L. Bruce Gibby, Ph. D.

*Jet Propulsion Laboratory (JPL), California Institute of Technology (Caltech)*

Randy Egger

*Project Partners, LLC*

### **Abstract**

Business units face special challenges when they must integrate commercial-off-the-shelf cost management software from one supplier with the enterprise financial system from another. Learn how the Jet Propulsion Laboratory integrated Cobra Cost Management software with Oracle Projects to build a powerful resource planning and management application used by cost estimators, project managers, task managers, and project resource administrators.

### **Executive Summary**

Most off-the-shelf enterprise financial systems do not come with robust cost management or project management capabilities. In order to do effective cost and/or project management, business units must develop “bolt on” tools or purchase third party project management software. Developing “bolt on” tools often involves customizing the enterprise financial system, which not only becomes costly and time consuming, but can lead to future upgrade and support problems. Purchasing third party software most likely saves time and money but can be difficult to interface to the company’s financial system. Some project management software vendors have partnered with suppliers of financial systems to solve the integration problems. However, the project management software is usually integrated with only one financial system. The impact is that—if you chose not to purchase systems from suppliers with such agreements—integration is difficult. To make matters worse, when software companies have teamed together, integration is often limited in capability. Oracle Corporation has taken a step forward by developing the Activity Management Gateway (AMG). The AMG is layer of programs designed to interface a variety of third party project management tools to Oracle Project Accounting. The AMG acts as a buffer between Oracle Project Accounting and the third party tools; thereby, reducing the issues related to cost, future upgrades, and support. This paper describes how the Jet Propulsion Laboratory teamed with Welcom, Project Partners LLC, and Oracle to use the AMG for integrating Welcom’s Cobra cost management/earned value management software with Oracle Project Accounting. This strategy produced an effective, enterprise-wide resource planning and management application for use by cost estimators, project managers, task managers, and project resource administrators.

### **Background**

#### ***About the Jet Propulsion Laboratory***

Jet Propulsion Laboratory (JPL) is a non-profit federally funded research and development center (FFRDC) located in Pasadena, California. JPL is operated under contract by the California Institute of Technology (Caltech) for the National Aeronautics and Space Administration (NASA). JPL is part of the United States aerospace industry. It is NASA’s premier center for unmanned exploration of the universe. Although NASA work predominates, JPL performs tasks for several other federal agencies such as the Department of Defense (DoD), the Department of Transportation (DoT), and the Department of Energy (DoE). JPL has roughly 5,100 employees:

4,200 in the technical divisions and 900 in the administrative divisions. JPL's annual budget is approximately \$1.4 billion.

***Scope of the Resource Planning and Management System (RP&MS) Task***

The scope of the RP&MS task included major business process reengineering for the areas of project management and project resource management. The RP&MS accommodates projects that employ earned value measurement techniques (EVM) as well as those that do not (Non-EVM projects). The technology that will enable these new processes consists of the following components:

1. Capable schedule tools that include, but are not limited to, MS Project from Microsoft and Open Plan Professional from Welcom Corporation.
2. JPL in-house developed, Microsoft Excel data entry spreadsheet with preliminary costing algorithms.
3. JPL in-house developed, MS FoxPro program that generates "flat" files from the Microsoft Excel data entry spreadsheet suitable for importing into Cobra.
4. Cobra Cost Management software purchased from Welcom Corporation.
5. JPL in-house developed Unix scripts, Oracle forms, and PL/SQL programs. This piece is known as the JPL middle ware.
6. Project Loader software purchased from Project Partners LLC.
7. Oracle Activity Management Gateway (AMG) purchased from Oracle.
8. Oracle Financials modules: Project Accounting (PA), General Ledger (GL), and HRMS.

The technical architecture of the RP&MS task had to interface each of these components. The focus of this paper is limited to the required interfaces between pieces four through eight.

***History of the Resource Planning and Management System (RP&MS) Task***

The late 1990s was an opportune time for significant change at JPL. With the advent of new NASA program and project management requirements (April 1997) and the new NASA prime contract (September 1998), the stage was set to conceive and implement a Resource Planning and Management System (RP&MS) with all around cost management and earned value capability. Earlier in the decade there were several attempts at implementing full criteria-based EVM systems. These attempts failed. Failure was attributed to two main causes:

1. Lack of well-defined project management processes.
2. Absence of integration between the project management system and JPL's enterprise financial system.

The earlier attempts occurred before the April 1997 and September 1998 events. At that time JPL was not fully supportive of implementing an integrated, full criteria-based EVM system. Even with the contractual requirements in the newly executed prime contract, senior management was not totally convinced that an integrated EVM system should be implemented at JPL. Not until the summer of 1999 was management certain that an integrated EVM system should and could be implemented at JPL.

**Scope of this Paper**

This paper will focus on the interfaces between Cobra and Oracle Financials. It will look at the technical aspects of that part of the RP&MS task. It will justify why interfaces are necessary,

describe the type of information that is shared, explain important factors shaping the design of the architecture, illustrate in technical terms this architecture, list the major challenges, and present the RP&MS task's most important lessons learned. The authors hope this paper will be of value to other companies that have similar objectives, requirements, and constraints.

## **Why Cobra Must Be Interfaced With Oracle Project Accounting**

### ***Budgets Created in Cobra Must Be Stored in Oracle Project Accounting***

Jet Propulsion Laboratory is required to provide NASA with a monthly report (NASA Form 533). This report displays actual cost incurred, estimate-to-complete (ETC), and estimate-at-completion (EAC) for every project funded by NASA. For three reasons JPL has mandated that the 533 report be produced from Oracle Project Accounting and not from the Cobra database:

1. Project Accounting is the source of all actual costs.
2. Project Accounting is directly updated with budgets from JPL's approved *custom budgeting tool*.
3. Reasons one and two have resulted in Oracle Project Accounting being the certified site for all JPL cost and budget data.

The second reason was the motive behind the upload architecture. Given that JPL's custom budget tool was implemented before Cobra was selected and procured, projects had no choice but to use that tool to create budgets in Oracle. By the time Cobra was purchased, numerous projects were creating budgets directly in Oracle. Furthermore, since using Cobra is not a requirement at JPL, many projects may continue to use the custom tool for preparing budgets. To avoid possible confusion by reporting budget information from two places (Cobra and Oracle), it was decided the best approach was to take budgets created in Cobra and find a way to store them in Oracle.

A serious shortcoming emerged with the decision to use Oracle Project Accounting as the source of the NASA 533 report. It was pointed out that JPL's custom budgeting tool was not designed to create ETCs/EACs. As mentioned earlier, estimates-to-complete and projected estimates-at-completion are required by Form 533; thus, all projects using the custom budgeting tool were unable to satisfy the Form 533 reporting requirements. Upon this discovery, JPL negotiated a special approach with NASA for reporting ETC and EAC projections. In the long run, one of three things is likely to occur:

1. JPL will continue to use the special approach for reporting ETCs and EACs.
2. JPL will modify its custom budgeting tool to include ETCs.
3. JPL will require all projects to use Cobra. Since Cobra creates ETCs/EACs, the custom budgeting tool may become obsolete, and the upload architecture will become unnecessary.

For now, JPL applies the special approach for NASA 533 reporting requirements. However, the wide spectrum in capability of the RP&MS suggests all projects may eventually convert to Cobra; thus, permitting Cobra to be the single source for all budget and EAC information.

### ***Costs Accumulated in Oracle Must Be Accessible to Cobra***

In order to compare costs against budget and to perform earned value analysis, Cobra must have access to actual costs. At JPL, all costs are recorded in Oracle Project Accounting. Therefore, two alternatives exist for Cobra having access to actual costs:

1. Cobra can access costs from Project Accounting on a real-time, transaction basis ("hot link").
2. Costs can be extracted from Project Accounting in text files and downloaded to Cobra.

### ***Obligation Amounts in Oracle Must Be Accessible to Cobra***

In order to compare obligated [committed] amounts to a project's funding profile, Cobra must have access to the obligation [encumbrance] amounts in Oracle Financials. Like costs, the same two alternatives exist for Cobra having access to obligations.

## **Type of Information That Must Be Shared Between Cobra and Oracle Financials**

1. Work Breakdown Structures
2. Approved Baseline Budgets
3. Estimate-to-Complete and Estimate-at-Completion Projections
4. Incurred Costs
5. Incurred Obligations [Encumbrances]
6. Global Information
  - a. Organization Breakdown Structure (Organization Chart)
  - b. Resource Breakdown Structure (Resource Types / Oracle Expenditure Types)
  - c. List of Employees (Managers)
  - d. Fiscal Calendar
  - e. Standard Labor Rates and Burden Rates
  - f. Oracle Project Accounting Task Attributes / Cobra Code Fields

## **Design Constraints**

Prior to 1999 all RP&MS team members were part-time, additional duty employees working in a "skunk works" mode (the importance of interfacing a project management tool to Oracle Project Accounting was not officially recognized by senior management until midway through 1999). Therefore, only limited money and time were allocated to building the prototype interface. Even worse, at the time this effort was started, Cobra had not been selected as JPL's enterprise project management tool. Given that any of several project management tools could be selected, the interface architecture had to be designed with a high degree of flexibility. Below are the major design constraints and the effects they had on shaping the overall architecture:

1. **Restricted cost and time** -- Rapid prototyping methodology had to be used. Management and users had to see a working model of the interface before funds would be provided. Also, existing hardware had to be used. There was neither enough time nor money to purchase high-powered database servers.
2. **A project management tool had yet to be selected** -- Interface architecture had to be flexible enough to work with an unknown tool. Furthermore, JPL management

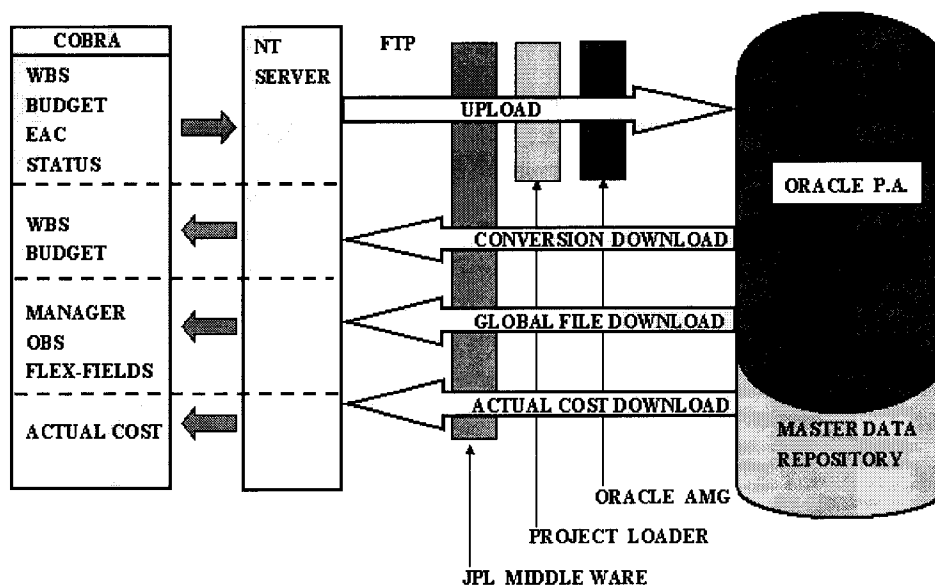
challenged the team to purchase an off-the-self project management tool and not develop one internally.

3. **Only store approved budgets in Project Accounting** -- JPL was experiencing severe disk space shortages with its Oracle Financials servers. To reduce the amount of space used, the Financial Controller strongly recommended that only approved budgets be stored in Project Accounting (it was not a requirement because JPL's custom budgeting tool already stored all of its budgets in Project Accounting: "what-if" budgets, budgets-in-progress, etc.)
4. **JPL's techno-culture** -- Users demanded sub-second response time. The response time and system availability they were experiencing with JPL's custom budgeting tool was deemed unacceptable. In order for funding to be approved, response time and system availability had to be far superior to what was currently being experienced with the custom tool.
5. **Store WBSs and budgets in Project Accounting during off shift hours** -- Because other Oracle applications, such as payroll, have higher priority, the RP&MS architecture had to be capable of limiting creation of WBSs and budgets in Oracle to off shift hours.
6. **Oracle's recommendation to use the Activity Management Gateway** -- Oracle recommends that all third party project management tools update Project Accounting using the Activity Management Gateway. The AMG is an application programming interface purchased from Oracle as a "bolt on" to Project Accounting.

### Architecture Design Solution

Working within the confines of the design constraints, the RP&MS team came up with an architecture that transfers "flat" files (TXT and CSV files) back and forth between Cobra and Oracle Project Accounting. Figure 1 portrays the conceptual block diagram of this architecture.

Figure 1 – Cobra to Oracle Interface Architecture



Admittedly, this approach is not glamorous or cutting edge, but was desirable for the following reasons:

1. **Restricted cost and time** -- At the time the RP&MS task was initiated, JPL was using two off-the-self project management tools from two different suppliers. Neither supplier was an Oracle CAI level partner, nor were they likely to become so in the near future. JPL contacted both suppliers; each agreed it would be cost and time prohibitive to develop transaction mode, hot linked interfaces between their tools and Project Accounting. In order to show JPL management a working prototype, Welcom built a WBS and budget flat file extract capability within JPL's cost and schedule constraints.

Existing hardware also had to be used. Experiences with JPL's custom budget tool and other Oracle modules led the team to believe that a high performance hot linked interface would require more robust hardware. Creating flat files and processing them in the background is not computer intensive. JPL's current hardware configuration was adequate for a flat file design.

2. **A project management tool had yet to be selected** -- As mentioned earlier, JPL was using two off-the-self project management tools. Although neither of the tools had been "officially" selected as JPL's enterprise tool, those tools—plus others—were considered for such use. In the event that JPL could not choose one official tool, the two existing tools may have to be interfaced with Project Accounting. However, in their "vanilla" modes, neither was capable of storing budgets directly in Project Accounting. JPL decided that the cheapest, fastest, and most flexible way to integrate cost and budget data between the tools and Project Accounting was to use flat file transfers. Flat file transfers meld nicely with both tools' standard import/export capabilities.
3. **Only store approved budgets in Project Accounting** -- Even though the two project management tools used at JPL created "what-if" and working budgets, only the approved baseline budgets and approved estimates-at-completion (EACs) should be stored in Oracle. The interface architecture had to plan for achieving this in either of two ways:
  - a. Limit the flat files produced by the project management tools to contain only approved baseline budgets and EACs.
  - b. If control could not be put into the project management tools, then select only approved baseline budgets and EACs from the flat files.
4. **JPL's techno-culture** -- JPL was experiencing poor response time with the hot linked custom budgeting tool. It was literally taking hours to create budgets in Oracle, with most of the time being spent waiting for the user's computer to return while Oracle processed transactions. The flat file design made it easy to use background Oracle processes. Background processing separates work done at the users' computers from that done on the Oracle servers; thus, eliminating the problem of "hanging" users' computers. Although flat file transfers are not required for background processing, they do fit nicely within that architecture.
5. **Store WBSs and budgets in Project Accounting during off shift hours** -- To control when WBSs and budgets are stored in Oracle, the RP&MS architecture first stores all updated WBSs and budgets in flat files on NT servers. Then either of two things happen:
  - a. Users can immediately upload their WBSs and budgets from the NT servers to Oracle.

- b. A nightly Oracle batch process can upload users' WBSs and budgets from the NT servers to Oracle.

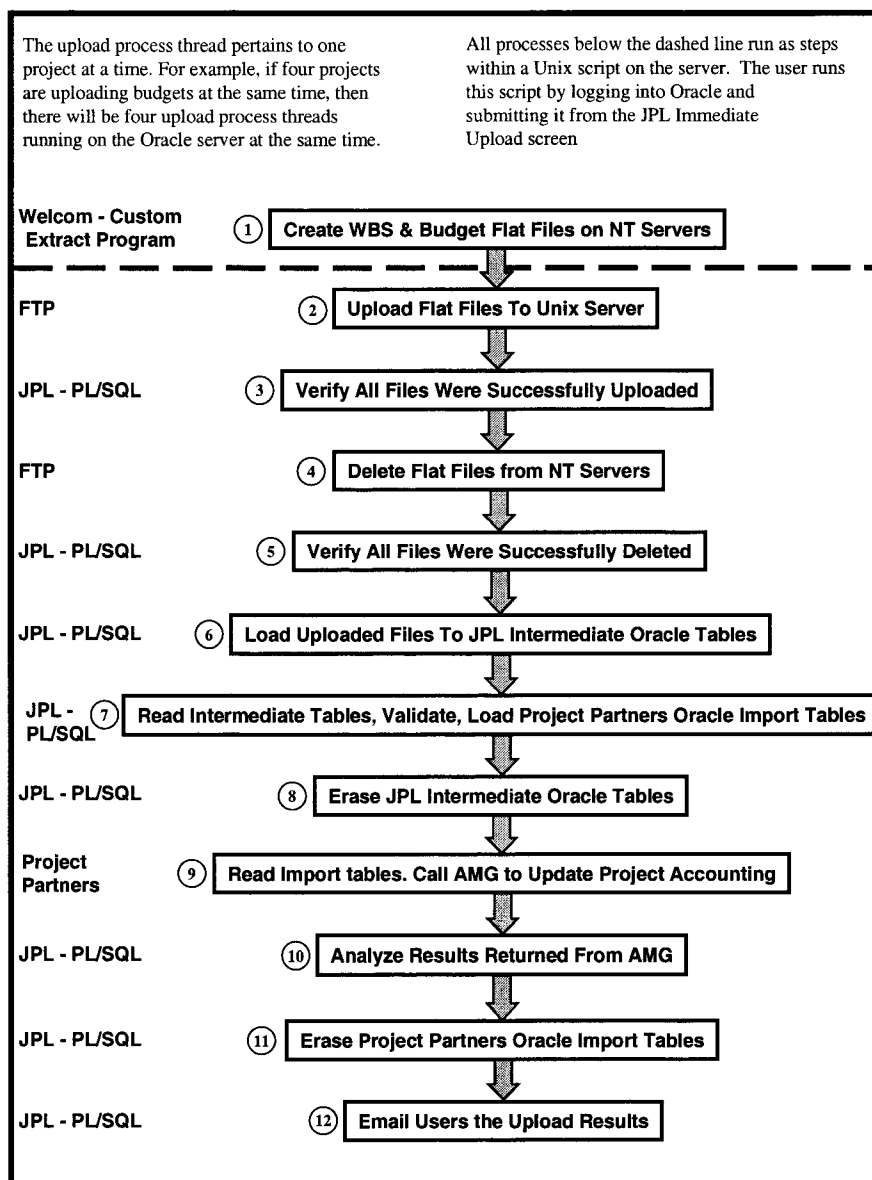
When Oracle processing is behind schedule, option one is turned off and only option two is in effect. This enables users to create WBSs and budgets during the day, save them on NT servers, and have them uploaded to Oracle, overnight, after all high priority jobs (i.e., payroll) have finished.

- 6. **Oracle's recommendation to use the Activity Management Gateway** -- Instead of directly updating the Project Accounting databases, JPL followed Oracle's recommendation and opted to use the Activity Management Gateway. To facilitate the use of the AMG, JPL purchased the Project Loader software from Project Partners LLC. The Project Loader does two things needed by JPL:
  - a. Simplifies using the AMG -- To speed up development time, the Project Loader provides Oracle interface tables. These tables are populated with uploaded WBSs and budgets. In JPL's case, the flat files produced by Cobra are uploaded to the Oracle environment and stored in these interface tables. The Project Loader reads these tables and performs the complicated logic of sending WBSs and budgets through the AMG to Project Accounting.
  - b. Insulates in-house developed Oracle programs from Project Accounting or AMG version upgrades -- When new versions of Project Accounting or the AMG are released, the Project Loader software is upgraded accordingly. In theory, changes will be handled internally in the Project Loader code, leaving the Project Loader interface tables untouched. Changes to JPL's custom programs should be minimized. This principle will be rigorously validated when JPL upgrades from version 10.7 to 11i of Oracle Financials.

### Upload Architecture

Figure 2 shows the major processing steps of the upload architecture. The upload process starts when a user has an approved baseline budget or estimate-at-completion in Cobra that is ready to be stored in Project Accounting. To initiate the upload, the user employs a custom Cobra form (see Figure 3) developed by Welcom for JPL. (This is the form that was mentioned earlier, as being developed within JPL's cost and time constraints.) The "Cobra Upload" executable creates flat files containing WBS and budget data on NT servers. If the user wants the data uploaded immediately, then the user must log into Oracle and submit a Unix script using the "JPL Immediate Upload" form (see Figure 5). The Unix script uses FTP to "pull" the flat files from NT servers to the Unix / Oracle Financials servers. A series of Oracle PL/SQL programs ultimately store the WBS and budget data in Project Accounting. **Note:** *Uploads are never "pushed" from the Cobra environment to the Oracle environment. They are always "pulled" from NT servers to the Unix / Oracle Financials servers.*

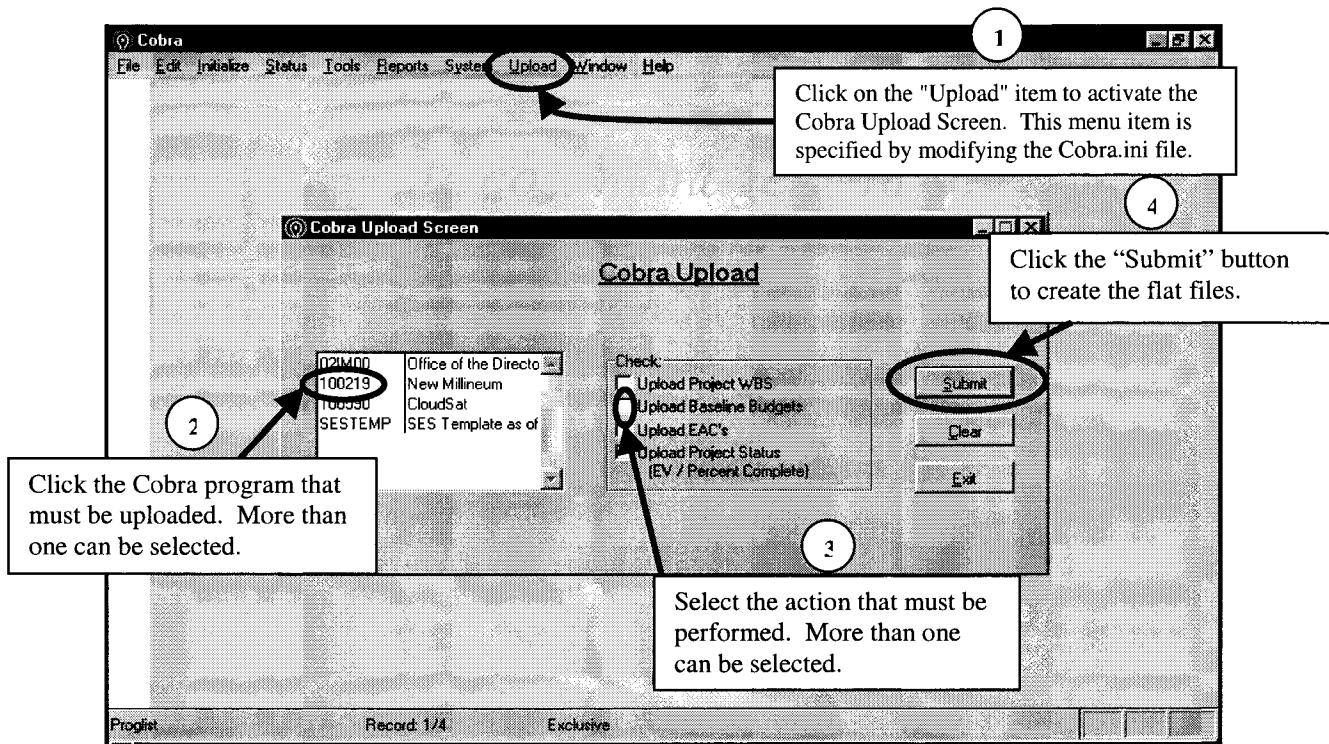
### Figure 2 -- Upload WBS and Budget Process



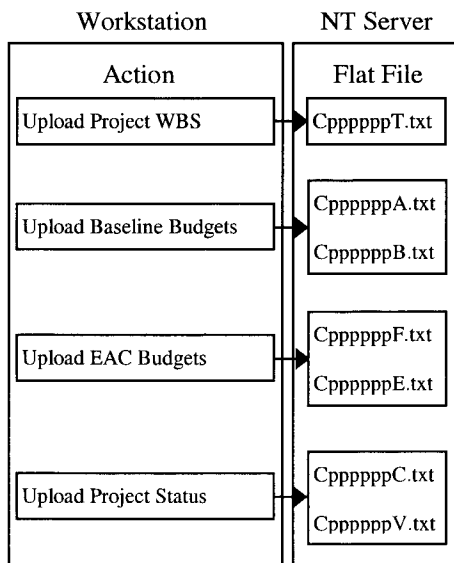
**Step 1 Create WBS and Budget Flat Files on NT Servers** -- In order to extract WBS and budget information from the Cobra databases, Cobra users apply the custom “Cobra Upload” form (Figure 3). Welcom developed this form from specifications supplied by JPL. Although the form and its related programs were developed using JPL specifications, other companies that use the Project Loader to interface with the AMG can use this form. The associated programs read the regular Cobra databases to retrieve WBS and budget information. The flat file extracts are virtually one-for-one matches with the Project Loader import tables.

Users access the Cobra Upload form directly from within the Cobra Cost Management software. If you are familiar with Cobra, you will notice (in Figure 3) along the menu one additional pull down item labeled “Upload.” Selecting “Upload” activates the form. Integration with the Cobra cost management software is achieved by two new entries in the Cobra.ini file. The first entry defines the new “Upload” item on the menu bar. The second entry specifies where to store the extracted flat files.



**Figure 3 --Cobra Upload Form**

The left part of the form lists every Cobra program that can be accessed by a given user. Click on as many programs as need to be uploaded; more than one program can be uploaded at a time. The middle part of the form lists the actions that can be performed. As few as one and as many as all the actions can be selected. The WBS is always uploaded, whether it is selected or not. By uploading the WBS, the system insures that the WBS and budgets being uploaded are synchronized. Notice that the uploads are for entire projects. Uploading parts of a project's WBS or budget are not allowed per JPL and AMG business rules. The "Submit" button is used to start the extract process. Figure 4 below shows the relationships between the upload action and the flat files that are produced.

**Figure 4 – Relationship Between Upload Action and Files Produced**

Flat file naming convention is as follows:

CppppppT.txt

- T = Task
- A = Baseline Budget Header
- B = Baseline Budget Details
- F = EAC Budget Header
- E = EAC Budget Details
- C = Percent Completes
- V = Earned Value

Six character project number. JPL chose six character project numbers as a business rule.

C = Cobra file. At the time this naming standard was developed, other project management tools were being considered for uploading.

Example 1: When the “Upload Project WBS” action is submitted for project 100234, file C100234T.txt is created on the NT server. Example 2: If “Upload Baseline Budgets” are submitted for projects 100976 and 100567, then files C100976A.txt, C100976B.txt, C100976T.txt, C100567A.txt, C100567B.txt, and C100567T.txt are created. Notice that the CppppppT.txt [WBS] file is always created, whether “Upload Project WBS” is selected or not. Also, all files remain on the NT server until they are deleted by the Unix / Oracle Financials server process.

**Step 2 Upload Flat Files to Unix / Oracle Server** – This is accomplished by means of FTP. A Unix script, run from the Unix / Oracle server, logs into the NT server, and “pulls” the flat files to the Unix server. To run this script, the must user log into Oracle and employ the JPL Immediate Upload form (see Figure 5). Only projects where the user is a “key member” appear on this form. A table in Oracle contains the NT server address, login name, password, and NT directory where the flat files are stored. When the user presses the “Submit” button , the form accesses this table and builds the necessary FTP control cards. It then submits the script as a concurrent process using Oracle concurrent manager.

### Figure 5 -- JPL Immediate Upload Form in Oracle

The screenshot shows the Oracle Applications interface for the Project Management Upload Request Form. The window title is "Oracle Applications - [Project Management Upload Request Form]". The menu bar includes Action, Edit, Query, Go, Folder, Special, Help, and Window. The toolbar contains various icons for file operations and navigation.

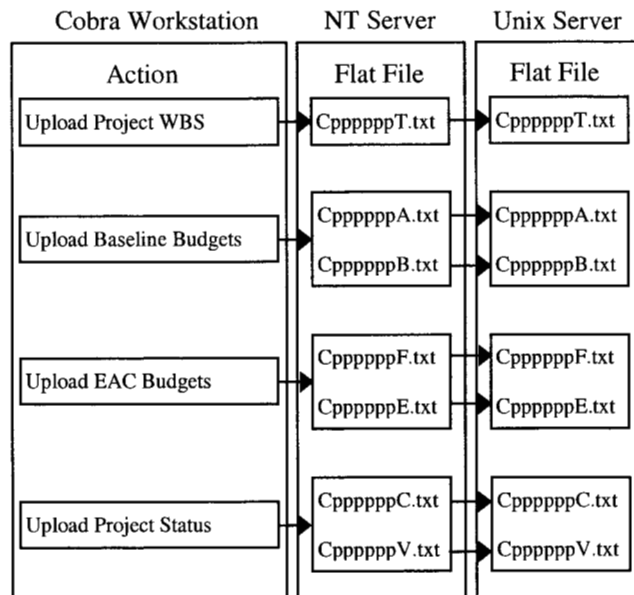
Action	Project Number	Project Name	Last Upload Date
<input type="checkbox"/>	02IM00	Office of the Controller	
<input checked="" type="checkbox"/>	100219	NEW MILLENNIUM PROG 10,	
<input checked="" type="checkbox"/>	100590	CLOUDSAT FORMULATION	
<input checked="" type="checkbox"/>	100673	MSR-Sample Transfer Chain	
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

A callout box labeled "1" points to the selected rows in the table, stating: "Click here to select the projects that must be uploaded. More than one project can be uploaded at a time."

A callout box labeled "2" points to the "Submit Upload Now" button at the bottom, stating: "Click the 'Submit' button to start the upload process."

The status bar at the bottom left indicates "Count: \*0".

When FTP "pulls" the flat files to Unix, it stores them in Unix using the same names they had on the NT server. Figure 6 on the next page traces the path of the flat files as they are uploaded to Unix.

**Figure 6 – Path of Flat Files As They Are Uploaded To Unix**

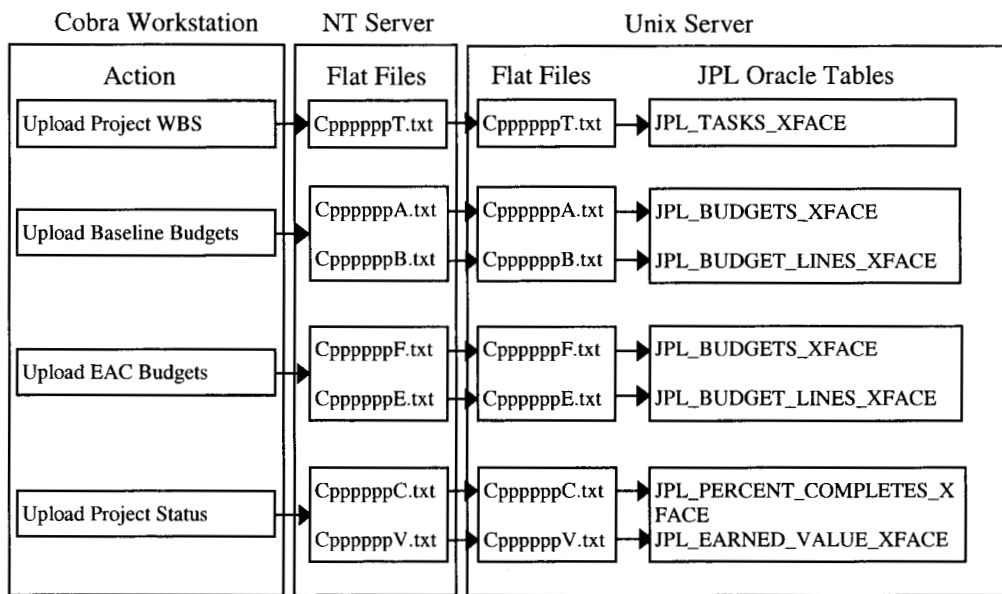
**Step 3 Verify all Files Were Successfully Uploaded** – The FTP upload process creates a log file on the Unix server. A JPL PL/SQL program reads this log file looking for errors. If there are errors, then e-mail messages with instructive diagnostics are sent to the user and application administrators, and the upload process is terminated.

**Step 4 Delete Flat Files from NT Server** – The FTP program is utilized again, this time to delete the flat files from the NT server. The flat files are deleted for two reasons:

1. Saves space on the NT server.
2. Ensures the same files are not re-uploaded when the regularly scheduled, nightly upload process is run.

**Step 5 Verify all Files Were Successfully Deleted** – Like the FTP upload process, the FTP delete process creates a log file on the Unix server. A JPL PL/SQL program reads this log file looking for errors. If there are errors, then e-mail messages are sent to the user and application administrators, and the upload process is terminated.

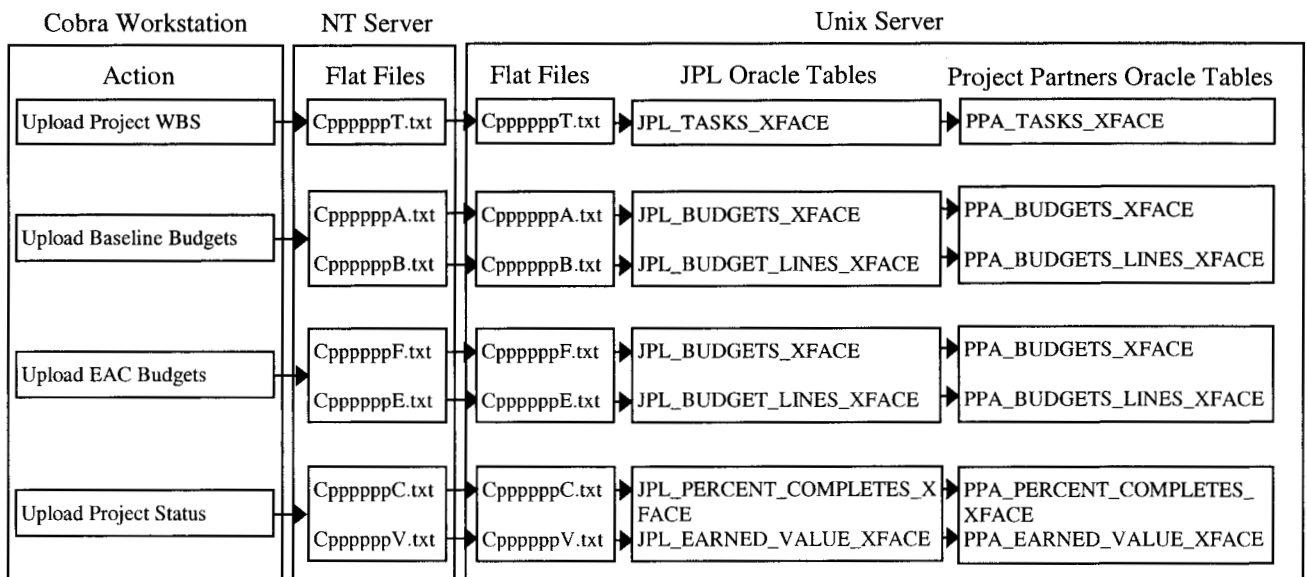
**Step 6 Load Uploaded Files to JPL Intermediate Oracle Tables** – The uploaded WBS and budget flat files are copied into Oracle tables. They are copied simply as a matter of convenience – it is easier to manipulate data in Oracle tables than in flat files. No JPL project management business rules are enforced here. Only simple validation of the flat files' formats is performed. Figure 7 on the next page shows the movement of data to the intermediate tables.

**Figure 7 – Progress of Data to the Intermediate Tables**

**Step 7 Read Intermediate Tables, Validate, Load Project Partners Oracle Import Tables –**  
Two important things happen during this step:

1. All JPL project management business rules are enforced.
2. The Project Partners import tables are populated.

The uploaded WBSs and budgets are read from the intermediate tables, rigorously validated, and written to the Project Partners import tables. All JPL specific business rules are enforced here. The rules are preset in the JPL PL/SQL programs. Any WBSs or budgets violating these rules are rejected with appropriate error messages. If there are rejections, then the entire upload is terminated. All messages are sent via e-mail to the users. Figure 8 below shows the progress of data as it moves to the Project Partners import tables.

**Figure 8 – Progress of Data to the Project Partners Tables**

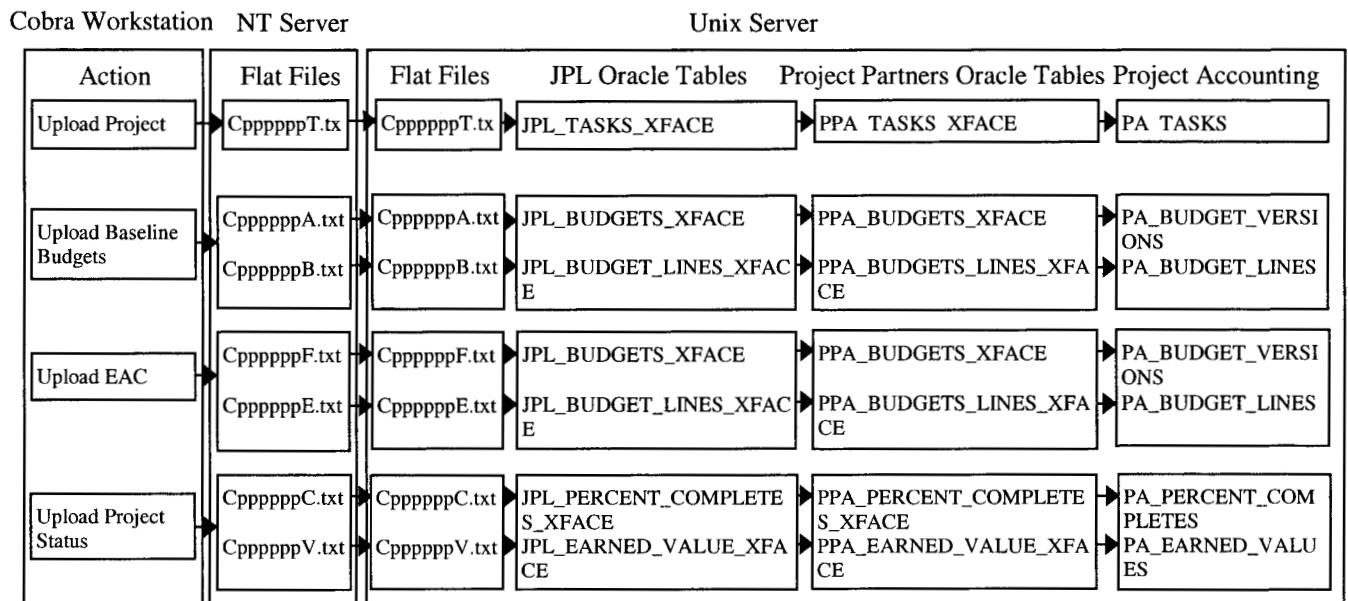
**Step 8 Erase All Intermediate Oracle Tables** – After the Project Partners import tables are populated, then the JPL intermediate tables are erased. The upload process no longer requires the intermediate data.

**Step 9 Read Import Tables and Call AMG to Update Project Accounting** -- The Project Partners software reads each import table and calls the AMG in order to update Project Accounting. The Project Partners software reformats this data into a format suitable for the AMG. It does not validate the data, all validation is done by the AMG. When updating a project's WBS, all records that do not pass the Project Accounting business rules are rejected. However, any records that pass the rules are accepted. This means that a project's WBS can be partially updated in Oracle; thereby, leaving the WBS in Cobra out-of-sync with the WBS in Project Accounting. When validating budget information, one reject causes the entire budget to reject.

The AMG returns all messages back to the Project Partners software. The Project Partners software stores these messages on the import tables in the same record that caused the message. Figure 9 shows the progress of data as it moves to Project Accounting.

**Important Note:** The messages returned by the AMG are very cryptic. Also, JPL has been unable to obtain any documentation from Oracle that helps explain these messages.

**Figure 9 – Progress of Data to Project Accounting**



**Step 10 Analyze Results Returned from AMG** -- After the Project Partners software updates the import tables with messages returned from the AMG, a JPL PL/SQL program reads these messages. For each record on the import table, it can be in one of three states:

1. Accepted without a message.
2. Accepted with a warning message.
3. Rejected with a rejection message.

The JPL program reads the messages and writes them to a custom Oracle messages history table. This history table is used in Step 12 to produce the e-mail file that is sent to the users.

**Step 11 Erase Project Partners Oracle Import Tables** -- After analyzing the results returned from the AMG, all Project Partners import tables are erased. They are always erased regardless of rejections, warnings, or no messages at all.

**Step 12 E-mail Users the Results of the Uploads** -- A JPL PL/SQL program reads the message history table. It extracts every message that was produced during the upload process. The extracted messages are written to a flat file on the Unix server. The Unix Mailx program is triggered which sends the flat file containing messages to the users. The users' e-mail addresses are obtained from the Oracle HRMS database. Users always receive e-mail messages, even if their uploads were successful.

### **Download Conversion Architecture**

The download conversion process converts projects presently in Project Accounting into Cobra projects. The download conversion accomplishes four things:

1. Edits and validates the project, WBS, and budget information in Project Accounting against JPL's and Cobra's business rules.
2. If the data in Project Accounting passes the validation, then the WBS and budget in Project Accounting are downloaded as .CSV files to NT servers. The downloaded data can be imported into Cobra.
3. After the data is downloaded, then the project's PM Product Code field in Project Accounting is updated with a value of "Cobra\_Vn.n", where n.n is the current version of Cobra.
4. Updates a custom Oracle table that controls whether a project can use JPL's custom budgeting tool to create budgets. The conversion process sets a switch that says the project can no longer use the custom budgeting tool.

The first item refers to validating WBS and budget information in Project Accounting. The first objective behind validation is to insure that the data will not reject when importing into Cobra. Cobra and Project Accounting have entirely different business rules. Some things that are acceptable in Project Accounting are not acceptable in Cobra, and vice-versa. Any piece of data that would reject when importing into Cobra is flagged with a rejection message. The download is terminated when rejections are encountered. All errors are e-mailed to the users. The users must correct the rejections in Project Accounting before their projects can be downloaded and converted.

The second objective of validation is to insure that, after data is successfully imported into Cobra, it will not reject during the upload process. Most commonly, if WBSs and budgets reject while uploading, it is because they failed to pass the JPL business rules. The download conversion issues warnings when the project, WBS, or budget fails any JPL business rules. (Indicates there is data currently in Project Accounting that violate JPL business rules. Why corrupt data enters Project Accounting is explained later.) The rationale for producing warnings instead of rejections is that the data will still import successfully into Cobra. However, if the information is not corrected in Cobra before uploading, then it may reject during the upload process.

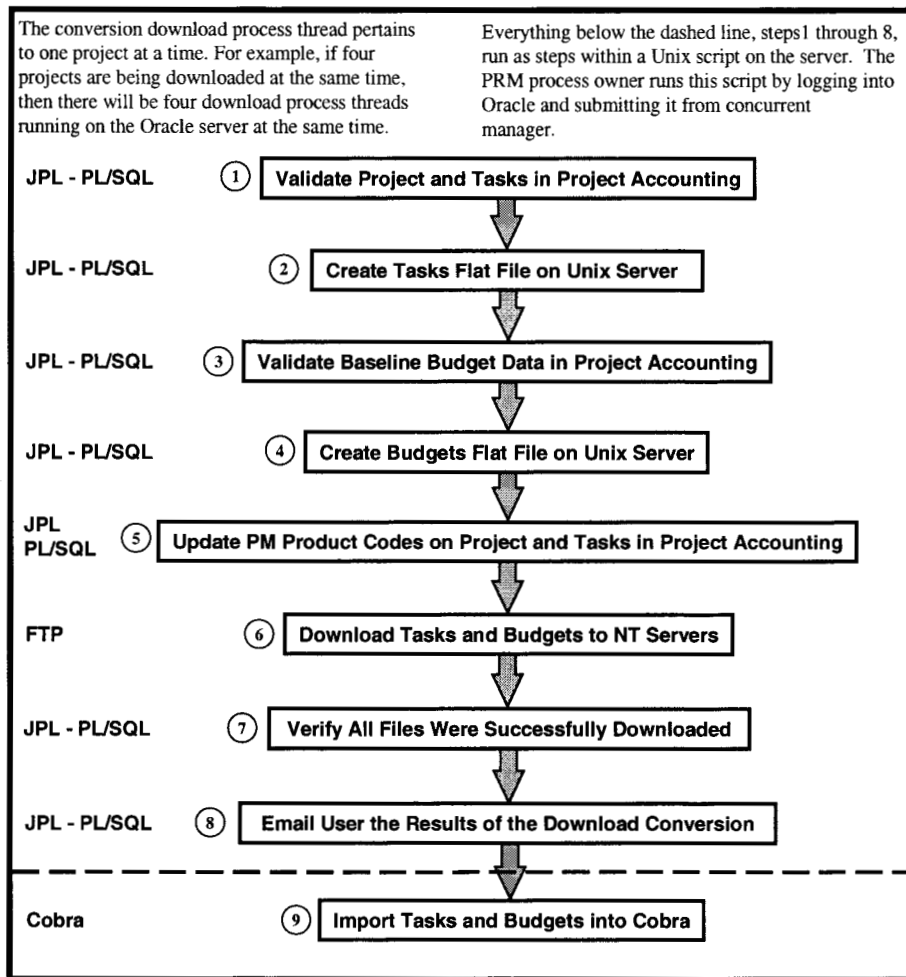
Item number two relieves the users from re-keying their WBS and budget information into Cobra. They simply import their downloaded files into the Cobra application. There are six files that are downloaded and imported:

1. CppppppT\_CONVERT.csv -- Cobra WBS file. Contains one record per project plus one record for each task belonging to the project. The tasks include parent tasks and cost accounts.
2. CppppppA\_CONVERT.csv -- Cobra activities file. Contains one record per work package. Work packages are not defined in Project Accounting; therefore, the conversion program defines them by categorizing budget amounts into fiscal year groups. The conversion program creates one work package per fiscal year per cost account.
3. CppppppCB\_CONVERT.csv -- Cobra cost class CB. Contains Labor, Travel, and In-house Services budget amounts.
4. CppppppCP\_CONVERT.csv -- Cobra cost class CP. Contains Procurements and Sub-contracts budget cost amounts.
5. CppppppOP\_CONVERT.csv -- Cobra cost class OP. Contains Procurements and Sub-contracts budget obligation amounts.
6. CppppppATTR\_CONVERT.csv -- Cobra code field values. Contains the flex field attributes from Project Accounting. There is one record per cost account. Even though Project Accounting allows flex field attributes at the parent task level, JPL is restricting Cobra to store these attributes at the cost account level only.

The third item refers to satisfying a requirement of the AMG. Before using the AMG to update a project, the AMG requires that the PM Product Code field for that project be populated with a value that identifies a third party tool. ***This is the only time that Project Accounting is updated without going through the AMG.*** This value must be stored in the Activity Management Gateway parameter of the Project Accounting setup. When updating a project's WBS or budget using the AMG, the AMG checks two things: First it checks the PM Product Code field for the project in Project Accounting to see if the project can be updated by a third party tool. Second, it checks the PM Product Code value on the uploaded transactions against the valid PM Product Code values in the Project Accounting setup. If all tests are passed, then the WBS and budgets can be updated using the AMG.

Item four ensures that budgets are created in only one place. If for some reason a project must use JPL's custom budgeting tool instead of Cobra to create budgets, then the switch is set to allow the custom tool and not allow Cobra to create budgets.

Figure 10 shows each step of the download conversion process.

**Figure 10 -- Download Conversion Process**

**Step 1 Validate Project and Tasks in Project Accounting --** As previously mentioned, the project and tasks records in Project Accounting are validated against Cobra and JPL business rules. The validation is done using a JPL PL/SQL program. Any change to the business rules requires a change to this program. As the data is validated, it is extracted from the Project Accounting database and written to flat files. These files are described in Step 2.

**Step 2 Create Task Flat File on Unix Server --** This step creates the CppppppT\_CONVERT.csv and CppppppATTR\_CONVERT.csv files on the Unix server. The six character project number is substituted for pppppp.

**Step 3 Validate Baseline Budget Data in Project Accounting --** A JPL PL/SQL program reads the project's latest approved baseline budget in Project Accounting. While it is being read, it is validated against Cobra and JPL business rules. If the budget is acceptable, then it is written to flat files on the Unix server.

**Step 4 Create Budget Flat Files on Unix Server --** This step creates the CppppppCB\_CONVERT.csv, CppppppCP\_CONVERT.csv, and CppppppOP\_CONVERT.csv files on the Unix server.



**Step 5 Update PM Product Code on Project and Tasks in Project Accounting --** As mentioned earlier, this is a requirement of the AMG in order to update project and task records in Project Accounting. The PM Product Code field is updated with the name and version number of the third party project management tool that will be using the AMG. This value must match one of the values in the Activity Management Gateway parameter list in Project Accounting.

**Step 6 Download Tasks and Budgets to NT Servers --** All six flat files are downloaded to the NT server. This is accomplished by means of FTP. A table in Oracle contains the NT server address, login name, password, and NT directory where the flat files must be stored.

**Step 7 Verify All Files Were Successfully Downloaded --** The FTP download process creates a log file on the Unix server. A JPL PL/SQL program reads this log file looking for errors. If there are errors, then e-mail messages are sent to the user and application administrators.

**Step 8 E-mail Users the Results of the Download Conversion --** A JPL PL/SQL program reads the message history table. It reads every message that was produced during the conversion process and stores them in a flat file in Unix. The Unix Mailx program is triggered, which reads the flat file of messages, and sends the messages to the users. The users' e-mail addresses are obtained from the Oracle HRMS database. Users always receive e-mail messages, even if their conversions did not produce errors.

**Step 9 Import Tasks and Budgets into Cobra --** The users receive e-mail messages that show the names and locations of their download files. The users make use of the regular Cobra import process to load these files into Cobra.

### **Download Global Files Architecture**

Global files refer to the resource breakdown structure, organizational breakdown structure, fiscal calendar, list of employees, standard labor rates, overhead rates, and optional code values. These pieces of information are common to all projects. They are stored in Cobra templates. The source of most of this information is Oracle Project Accounting.

Using Project Accounting as the source requires that this data be downloaded from Oracle and imported into Cobra templates. To ensure that Cobra is as current as Project Accounting, triggers have been created in Oracle that immediately send e-mail messages to the Project Resource Management process owner when this information changes in Project Accounting. The e-mail messages instruct the process owner to submit the appropriate global download jobs. When possible, the triggers submit download jobs for the process owner. An exception to this method is the resource breakdown structure. For two reasons, JPL has chosen not to download the Project Accounting resource types:

1. Project Accounting does not have a resource breakdown structure similar to the one in Cobra.
2. There are resources in Project Accounting that JPL does not want in Cobra.

Project Accounting has only a two level resource breakdown structure. It has resource categories and resource types. JPL has a five level resource breakdown structure in Cobra. When a new resource is added to Project Accounting, the Project Resource Management process owner receives an e-mail describing the new resource. The process owner manually adds the new resource into its correct position in the Cobra resource hierarchy.

Also, there are several resources in Project Accounting that JPL does not want in Cobra. These resources are undesirable for two reasons:

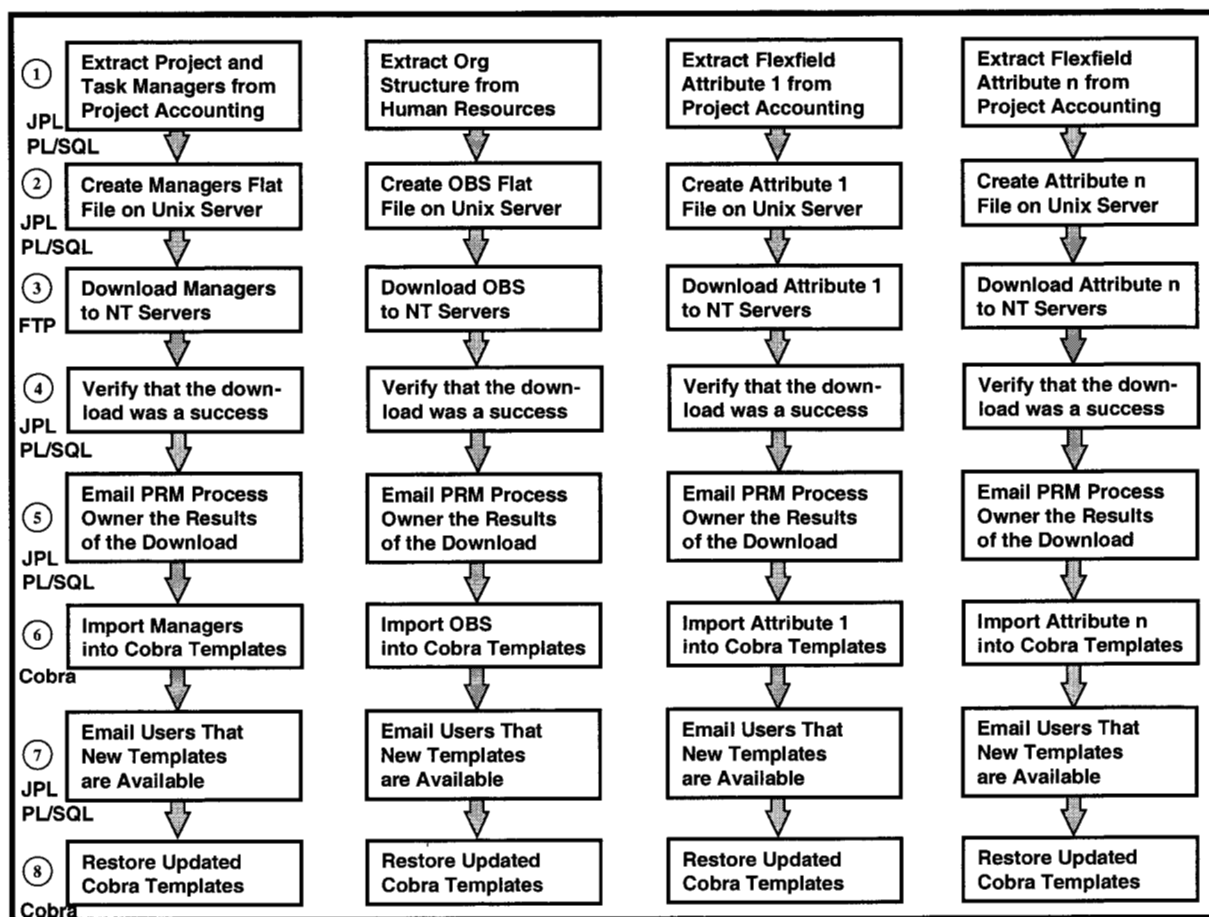
1. Resources are at a minute level of detail. A low level of detail is required by certain JPL internal and Caltech accounting needs. It is not desirable for project management purposes. Sales taxes are a good example.
2. Adjustments or one-time costs. Year-end adjustments are an example where costs may be incurred but not budgeted.

JPL has built a resource mapping table in Oracle that maps low level and one-time Project Accounting resources to higher level Cobra resources.

Worth noting is the list of employees. Only project and task managers are downloaded and imported into the Cobra list of employees. To determine which employees are project and task managers, a JPL PL/SQL program examines every project and task in Project Accounting and selects those employees that are assigned as project and task managers. Out of the 5,100 JPL employees, roughly 1,700 have at one time been a project or WBS task manager.

Figure 11 shows each step of the global file download process.

**Figure 11 -- Download Global Files**



### **Download Actual Costs and Obligations Architecture**

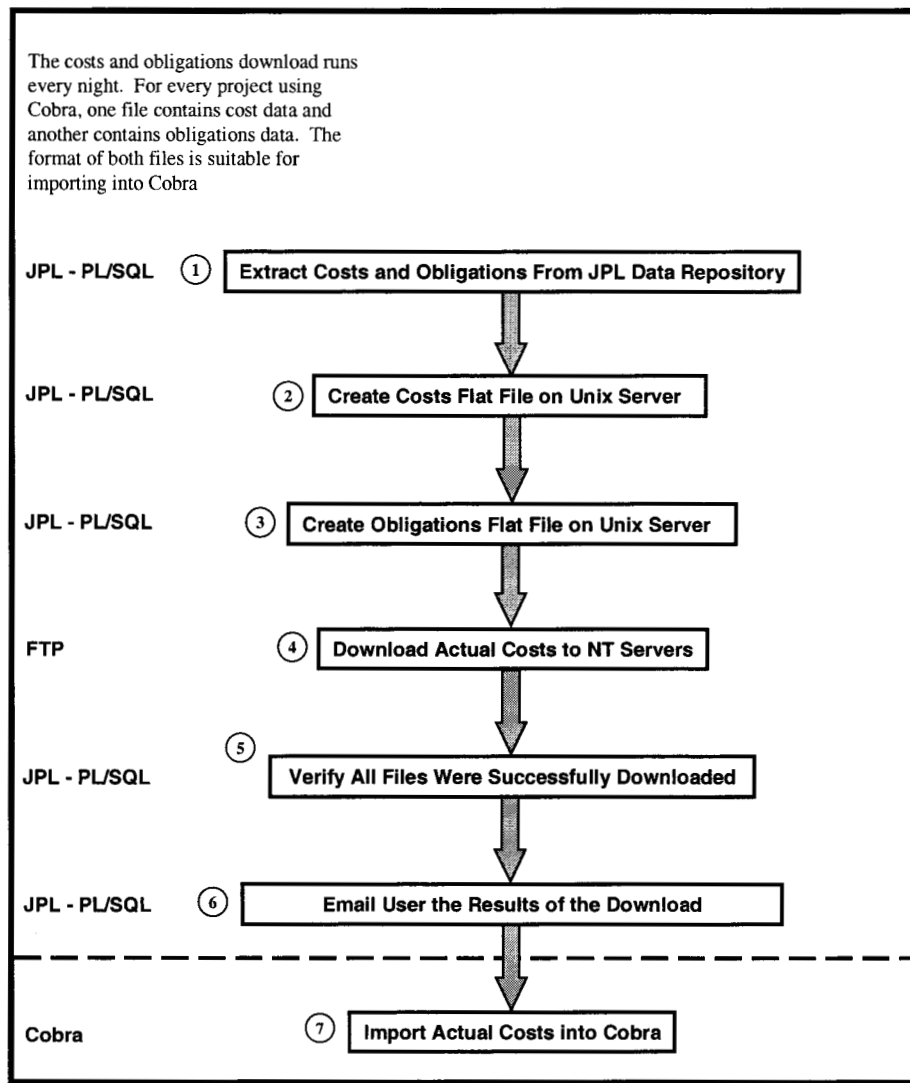
Actual costs and obligations are extracted from Oracle and downloaded to Cobra. Earlier it was mentioned this information was obtained from Project Accounting. *That is not entirely true.* JPL has created a data repository containing costs and obligations copied from Project Accounting. This repository is a set of de-normalized Oracle tables containing an assortment of data from Oracle Financials. The data is summarized in monthly intervals by project, task, and expenditure (resource) type.

The processing steps of the costs and obligations download are the same as the other download capabilities. However, the costs and obligations download runs every night; whereas, the other download jobs run only when needed. Because the information in the data repository is in monthly intervals, the download job usually runs without downloading anything. Only after the month-end books are closed does the job download data. After the files are downloaded, the users receive e-mail messages alerting them that their files are downloaded and ready to be imported into Cobra.

The download capability offers a current period download and an inception-to-date download. Users have a choice as to which type they would like to keep in Cobra. The file naming convention is as follows:

1. CppppppC\_CP\_MONYYYYY.CSV – Cobra current period cost file. This file contains current month costs. It is summarized by project, task, and expenditure type.
2. CppppppC\_ITD\_MONYYYYY.CSV -- Cobra inception-to-date cost file. This file contains inception-to-date costs updated every month. It is summarized by project, task, and expenditure type.
3. CppppppO\_CP\_MONYYYYY.CSV – Cobra current period obligations file. This file contains current month obligations. It is summarized by project, task, and expenditure type.
4. CppppppO\_ITD\_MONYYYYY.CSV -- Cobra inception-to-date obligations file. This file contains inception-to-date obligations updated every month. It is summarized by project, task, and expenditure type.

Figure 12 on the next page shows each step of the costs and obligations download job.

**Figure 12 -- Actual Costs Download From Project Accounting**

## Challenges Facing The Design Team

### *Nature of Task Start and End Dates in Project Accounting*

This issue has to do with the manner in which Project Accounting uses task start and end dates. Initially we assumed that the baseline start and finish dates from Cobra cost accounts would work nicely in Project Accounting as task start and end dates. However, we soon realized that these dates have different meanings between the two systems. In Project Accounting, task start and end dates are used to control when costs can be charged to a task. In Cobra, task start and end dates are used to schedule when work should be done on a task. For example, when all work on a task is finished, the project resource administrator will status the Cobra cost account finish date. When this date is uploaded to Project Accounting as the task end date, all manner of problems originate when there are "trickle" charges to the task. The charges are booked to a "suspense" account. The accounting staff has to determine why charges went into suspense, and then make correcting entries to resolve the problem. To avoid these problems, the design team chose not to upload task end dates from Cobra to Project Accounting. Instead, all task end dates will be managed directly in Project Accounting.

### ***Cleaning Up Corrupted Data***

Data corruption existed in the Project Accounting database. This situation occurred primarily when JPL converted from its old legacy system to Oracle Financials. The corrupted legacy data was not cleaned up prior to the conversion. When converting the legacy data, JPL used PL/SQL programs, thus bypassing all Project Accounting validation checks. Furthermore, some new data has been corrupted as it has entered Project Accounting. Oracle's lenient enforcement of standard industry project management rules and improper setup of certain Project Accounting parameters can result, among other things, in: 1) cost accounts being associated with invalid organizations, 2) blank project and task start dates, 3) erroneous project task start dates, 4) budget amounts occurring before task start dates. The design team developed rigorous logic in the download conversion program to handle this invalid data.

### ***Real-Time User Communications***

Because the upload architecture makes heavy use of background processing, the user community was concerned about timely feedback regarding the results of their uploads. To guarantee timely results, the team devised an innovative e-mail alert capability. The system uses the Unix Mailx program to send e-mail to users. A custom Oracle table stores all messages while they are being produced throughout each step of the upload. At the end of the upload, a JPL PL/SQL program reads the custom table; thereby, extracting all messages that happened during the upload. These messages are written to a text file on the Unix server. The Mailx program sends this text file to the user in the form of an e-mail message.

There are several benefits to this type of user communication. First, the user can retain the e-mail message for future reference. Second, the user can forward the e-mail to the Help desk. Third, all errors are bundled so that they can be corrected at one time. Fourth, certain messages can be automatically sent to the Project Resource Management process owner or application support staff. Lastly, hypertext links can be embedded in the messages that will take users to web-based training scenarios that are applicable to the problems encountered. The text for the error messages is stored on the Oracle Financials message table. The text can be edited for special or one-of-a-kind communication without changing any programs.

### ***Two Possible Sources for WBS and Budget Data***

With the advent of the RP&M system there emerged two sources for updating WBSs and budgets in Project Accounting. For WBS data, the first source was the Project Accounting forms. For budget data, the first was JPL's custom budgeting tool. With the introduction of the upload architecture, Cobra became the second source from updating WBSs and budgets. Obviously this had to be controlled. To control when, where, and how WBSs and budgets could be updated in Project Accounting, a custom Oracle table was created that stores such parameters. When a project is converted from Project Accounting to Cobra, the conversion process sets a "switch" on this table; thereby, blocking any further budget inputs from the custom budgeting tool. (The custom budgeting tool was modified to honor this switch.) However, WBS information can still be updated from Project Accounting and Cobra. This problem is not completely solved. For now, the upload from Cobra will always overlay certain fields in Project Accounting; thereby, making the WBS in Cobra the same as the WBS in Project Accounting. Training and ease of use of Cobra will probably result in the users using Cobra instead of Project Accounting to update their WBSs.

### ***Immediate Notification When Global Files Change In Oracle***

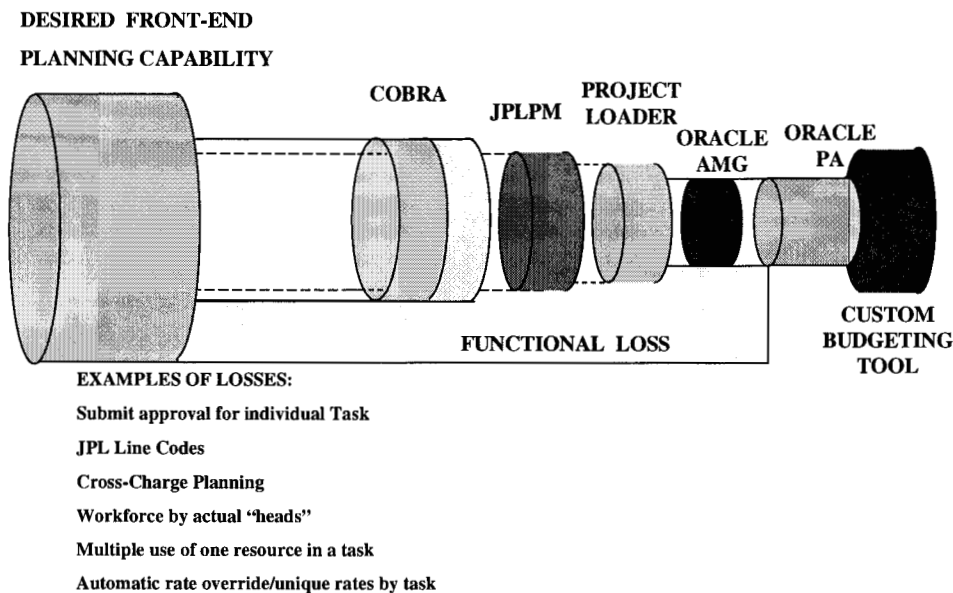
As described in the global file download architecture, it is important to keep the Cobra templates updated with the most current information from Oracle. To achieve this, a trigger mechanism was developed that sends e-mail messages to the Project Resource Management process owner. When changes occur in Oracle (resource types, organization structure, flex fields, etc.), database triggers "fire off" concurrent processes that send e-mail messages. In some cases, where review

of the e-mail messages is not required prior to downloading, the trigger submits the download job.

### **Functional Loss When Integrating COTS Software**

Off-the-shelf software is designed to meet the needs of a wide array of customers. Generally, third party software meets standard industry requirements. Often, users of this software benefit because it forces them into adopting standard industry processes. Equally as often companies suffer because they are unable to adapt to standard practices. In the case of JPL, a great deal of JPL specific functionality loss occurred when integrating Cobra with Project Accounting. Figure 12 illustrates the functional loss of integrating each layer of software.

**Figure 12 - The Impact of Software Constraints**



The challenge here is getting the user community to accept the functional losses associated with integrating COTS software. To do that, the losses must be proven to have marginal value. Educating users on the benefits of adopting industry standard project management techniques is critical to the success of the upload and download architecture.

### **Lessons Learned**

1. Clean up the data prior to system design: JPL expended significant effort designing around corrupt data—on a task that was resource constrained from the beginning. It is recommended that data cleanup be planned and budgeted for in the transition to a new enterprise accounting system. If this *is* accomplished, the design of the system will not be burdened with accommodating bad data. If this *is not* done, the resulting design may have areas that are inefficient or even problematic.
2. Obtain company acceptance of new business rules prior to system design: Business rules were defined *before* the architecture was designed. However, the JPL user community had not accepted them. The RP&MS task team assumed that these rules would be accepted, and therefore, designed the system with that in mind. Much re-programming was done to "soften" the original, strict business rules.

3. Select all COTS software prior to development: The architecture had to be flexible enough to work with an unknown project management tool. If the tool had been selected prior to system design, the design may have taken on a different flavor (some tools are partially integrated with Project Accounting).

## **Summary**

The RP&M System provides integrated processes with an enabling technology for passing cost and schedule data from Cobra to Oracle Project Accounting. The processes and technology are used by teams throughout JPL to manage their projects. By using as much off-the-self software as possible, and teaming with the suppliers of that software, a small development team was able to implement the system for relatively low cost. Admittedly, the resulting FTP "flat" file architecture is technologically archaic, but it meets or exceeds its performance expectations. Other companies with similar needs and resource constraints should consider using this technology. In time, the RP&MS team will look at other technologies with the potential to replace the "flat" file approach. Nonetheless, the latest and greatest technology is not always the best for a particular situation. New technology will replace the existing one only if superior performance versus the current RP&M System is evident.

## About the Authors

**J. Steven Joines** served as software engineer and lead developer for the RP&MS team during the design phase, development phase, implementation phase, and transition to production. Steve has 22 years of experience in implementing business information systems and Information Systems management. For the past three years he has worked as a Business Analyst and software engineer on Oracle Project Accounting. Steve obtained his B.S. degree from California State Polytechnic University, Pomona and is completing his M.B.A. from the same institution.

**L. Bruce Gibby** served as the process engineer and functional architect for the RP&MS team during initial concept phase, design phase, development phase, implementation phase, and transition to production. He has extensive industrial and consulting experience in project controls, project management, and business systems development. He has worked for two years on the integration of third party project management tools with Oracle Projects at JPL. He holds a B.A. in Political Science from Occidental College, an M.B.A. in Finance from the University of California at Los Angeles, and a Ph. D. in International Business and Comparative Management from the University of California at Los Angeles. Bruce was a Howard Hughes Staff Fellow and NDEA Title IV Fellow during his graduate studies at UCLA.

**Randy Egger** is the Founder and President of Project Partners, LLC. Previously Randy was the Chief Architect and Director of Oracle Projects at Oracle Corporation and has been in the applications business since 1980.

## Acknowledgments

The presenters gratefully acknowledge the contribution of the following individuals: Cal Chambers, the Project Resource Management process owner and RP&MS task leader, for his leadership, supervision, and mentoring support; Carolyn T. Stevens, manager of the Project Resource Administration Division, for recognizing the need for RP&MS and rallying institutional support; Anita Ho for her efforts in Software Development; Bredt Martin, for his EVM critiques; and Woody Lott of Welcom for his efforts in Cobra software implementation support.

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology under contracts with the National Aeronautics and Space Administration, NAS7-1260 and NAS7-1407. Reference herein to any specific commercial product, process or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.